An important characteristic of Knowbot programs is their ability to protect data on behalf of rights holders, to control its application in accordance with stated terms and conditions and to interpret data, rendering it in multiple ways to be visible, audible or otherwise palpable and in such a way as to permit human interactions to influence the choice of renderings.

The Knowbot framework defines both a "currency" for having tasks of any arbitrary complexity done, and a "medium" for handling the currency as part of getting the tasks done. The currency is called Knowbot programs. A Knowbot program is a mobile emissary of a network participant which assists in executing a task to be done on behalf of the participant. This task may be carried out locally in the user's system or it may involve interactions with other systems and resources at other locations both local and/or remote. The medium for handling Knowbot programs is called a distributed Knowbot service environment or simply the Knowbot service environment. The Knowbot service environment is distributed as a potentially endless number and variety of what we call Knowbot service stations (software and/or hardware) operating at places on the network. The stations may generate, store, execute and delete Knowbot programs and otherwise perform all allowed operations on Knowbot programs as determined by the Knowbot program itself or by users authorized to take or enable such actions. The Knowbot framework defines a minimal set of rules 38 for the structure of Knowbot programs (for example, the program must include some information related to the task to be done) and a minimal set of rules 40 for the functions to be performed within the Knowbot service environment (for example, that a typical Knowbot service station be able to create, send, receive, monitor, and delete Knowbot programs).

A participant at a computer (e.g., a person or a process or a Knowbot program running on the computer) may define the task to be executed. Information related to the defined task may be embedded in one of the Knowbot programs. Task information contained in Knowbot programs may be interpreted. The step of advancing the execution of tasks may include: creating additional Knowbot programs, interacting with other Knowbot programs, interacting with repositories of digital objects; querying a database of information; applying stored expert knowledge; protocol transformation; providing a directory service identifying other locations which provide services related to the task; making a security determination; providing a "what's new" service to identify newly available information; providing a "clipping" service which extracts information; providing a version control service for managing versions of the Knowbot program; responding to actions of a user by obtaining execution of the task at another location without necessarily indicating to the user that the execution occurred at the other location; providing a method for determining the structure of a Knowbot program (e.g., to determine if a digital object is contained in the Knowbot program and to access the element; creating a derivative work from an existing work; generating a Knowbot program that contains another Knowbot program, or a digital object, or other data; and passing a message from a source Knowbot program to a target Knowbot program at a different location.

The information concerning a task to be done may include interpretable or executable instructions. The Knowbot program may include data in the form of a digital object or a Knowbot program (a Knowbot program may itself be a digital object). The digital object may include protocol transformation information.

Knowbot Service Stations

The Knowbot service station is analogous to an operating system but does not allow participants to create arbitrary programs and files or to run arbitrary programs, although it could allow selected users with special system status great flexibility to manage the service station and to upgrade its performance. In general, the service station creates Knowbot programs in response to participant specifications and/or its own internal requirements. Prospective Knowbot programs may also arrive from external sources such as a user's PC that is not also a service station and can be authenticated by a service station for use in the Knowbot environment.

Prospective Knowbot programs may also arrive on external media such as CD-ROM. The service station also processes Knowbot programs received from other service stations, and stores and manages Knowbot programs within its own environment.

As seen in Figure 4, the functions performed at each service station may generally fall into four major categories: i. basic administration 60 of Knowbot programs, include creating, sending, receiving, authenticating, executing, storing, monitoring and deleting them; ii. interaction 62 with a participant 63 to aid the participant in defining tasks to be done and to provide the results of doing tasks in forms and at times that are useful; iii. interaction 64 with a participant which is serving as a resource 65 in getting a task done, including conveying requests for information or actions in formats and at times that will be understood by the resource and processing the information or results of the actions; and iv. interacting with other service stations 66 and interacting with hardware and software that is not part of the Knowbot framework to enable a limited set of external, possibly untrusted actions when appropriate and allowable.

Knowbot programs 54e through 54p may pass from Knowbot service station to Knowbot service station in the Knowbot service environment by a variety of mechanisms, for example by being transported in Internet packets. Knowbot programs 54q through 54t also may stay for periods of time in a given Knowbot service station and in that sense the host Knowbot service station may be thought of as a hotel and/or processing plant for Knowbot programs.

Referring to Figure 4B, the general structure of a generic Knowbot service station includes storage 60 for Knowbot programs and Knowbot program interpreters 62 for interpreting Knowbot programs written in languages such as (but not limited to) PERL or PYTHON. A core Knowbot operating system 68 provides basic key functions (discussed below) and provides a Knowbot operating system application program interface (API) 64 which provides a predefined interface to and thus enables application programs to be easily written to interact with the operating system. A Knowbot operating system extensions API 66 provides an interface to extensions 70. The extensions can include an almost limitless variety of services and functions that supplement the basic functions of the operating system, such as natural language processing.

Simple Knowbot service stations need only be able to perform a small set of basic functions and they may be implemented easily and in a straightforward way in the form of a software program or VLSI chip. Other more complex Knowbot service stations will have a variety of special functions suitable to their locations in the network and to the entities with which they are expected to interact.

We turn to a discussion of a simple Knowbot service station with minimum basic functionality. As shown in Figure 5, the heart of the Knowbot service station is the transaction manager 70 which acts as a kind of traffic cop and status handler to supervise activities within the service station. The transaction manager interacts with a Knowbot program receiver 72 where Knowbot programs arrive at the service station, and with a Knowbot program transmitter 74 where outgoing Knowbot programs depart the service station. An input scheduler 76 keeps track of the arrival of Knowbot programs and assures that they are scheduled for handling in the service station at the appropriate time. An output scheduler 78 similarly keeps track of Knowbot programs that are to be sent out of the service station and assures that they are transmitted in due course.

Knowbot programs and Knowbot service stations within the Knowbot service environment may be designated as "qualified entities," that provide certain assurances on system behavior although such assurances may not be possible for all service stations. Various levels of assurance may also be designated. For a service station, qualification amounts to registering the existence of the service station within the Knowbot service environment. The party applying for such registration would agree not to modify (or otherwise tamper with) the service station except as authorized. The service station could contain mechanisms to determine its own integrity (e.g., if it has been altered or its operation otherwise modified).

Knowbot programs arriving at a qualified service station may be allowed to be handled there without regard for the qualification of the service station which originated the program, provided the cost of execution is acceptable. For executions that are costly, electronic payment may be required to be authorized or included. If digital objects subject to rights (e.g., computer programs) or relating to rights (e.g., contracts or deeds) are requested for access, the Knowbot program may only be allowed to be returned to qualified service stations at a certain level of trust. A test may be made prior to performing the access to determine if the source service station of the program is qualified or not. Or, in a less restrictive scheme, the program may simply be limited in what can be done with the digital object.

For Knowbot programs, qualification may mean simply that a qualified service station was the creator of the program, assigned it a globally unique identifier, and maintains a copy of the Knowbot program along with any associated information. When a Knowbot program is deleted, a status message is returned to the service station that created it. This message may be deferred in delivery if that station is not available. Other mechanisms for notification in the event of long-lived Knowbot programs and relatively short-lived stations involve transfer of responsibility by notification to other stations.

A Knowbot program could be used to protect sensitive or confidential information. In this case, the terms and conditions might be based on the identity of the user and his need to know. Knowbot service stations could include third-party value-added service providers that facilitate the transfer of information from information providers to information consumers.--

In the claims

1. (previously amended) A method for use in a distributed system for processing a mobile program that has the ability to move from node to node in the distributed system comprising